

# Übungen zu Virtuelle Maschinen, Aufgabe 3

Volkmar Sieh

WS2022/2023

## 1 Übersicht

In Aufgabe 3 soll die bisher entwickelte virtuelle Maschine um Interrupts und einen Timer erweitert werden.

## 2 Organisatorisches

Bitte senden Sie Ihre Lösung bis zum 13.01.2023 per Mail an [i4vm@cs.fau.de](mailto:i4vm@cs.fau.de). Die Projektdateien mit Test-Programmen finden Sie unter <https://sys.cs.fau.de/lehre/ws/vm/uebung>.

## 3 CPU

Wird der CPU ein Interrupt „A“ gemeldet und ist das „I“-Bit des Statusregisters gesetzt, legt die CPU ihren aktuellen Programmzähler auf dem Stack ab, setzt das „I“-Bit auf „0“ und setzt den Programmzähler auf den Wert „A\*2“. Dort wird die Programmausführung fortgesetzt.

Die folgenden Interrupts sollen implementiert werden:

**1 (INT0):** Signaländerung an Pin 2, Port D

**2 (INT1):** Signaländerung an Pin 3, Port D

**7 (TIMER1 COMPA):** Ablauf Timer 1

Um Interrupts sinnvoll verwenden zu können, braucht man zusätzliche Instruktionen: `cli` (Interrupts blockieren), `sei` (Interrupts deblockieren) und `sleep` (CPU bis zum nächsten Interrupt anhalten).

Hinweis: wird das „I“-Bit auf „1“ gesetzt, werden die Interrupts erst nach der *nächsten* Instruktion deblockiert.

## 4 Interrupt-Leitungen

Der ATmega32-Mikrokontroller kennt drei Leitungen, auf deren Signaländerungen er mit Interrupts reagieren kann. Zwei davon sind zu implementieren: INT0, INT1.

Diese Interrupts können mit Bit 6 (INT0) und Bit 7 (INT1) des I/O-Registers GICR (Adresse 0x3b) aktiviert bzw. deaktiviert werden. Bei welcher Signal-Flanke ein Interrupt ausgelöst wird, kann über I/O-Register MCUCR (Adresse 0x35) eingestellt werden. Ob ein Interrupt „pending“ ist, wird über die Bits im Register GIFR (Adresse 0x3a) der CPU mitgeteilt.

## 5 Timer

Das Testprogramm verwendet nur Timer „1“ (16-Bit-Timer) des Mikrokontrollers. Nur dieser muss implementiert werden. Vom Testprogramm nicht verwendete Features dürfen weggelassen werden. Z.B. wird nur der Modus „Clear Timer on Compare“ („CTC“; Modus 4) verwendet.

Der Timer wird über die I/O-Register TIMSK (Adresse 0x39), TIFR (Adresse 0x38) sowie die Register TCCR1A, TCCR1B, TCNT1H, TCNT1L, OCR1AH, OCR1AL, OCR1BH, OCR1BL, ICR1H und ICR1L (Adressen 0x26 bis 0x2f) programmiert bzw. ausgelesen.

## 6 Testprogramm

Auf der Übungsseite finden Sie das Testprogramm „boardtest“. Ihre VM soll in der Lage sein, dieses korrekt abzuarbeiten.